UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/911,819	07/24/2001	John T. Micco	04899-046001	6291

7590 05/18/2007

Kevin J. Canning, Esq.
Lahive & Cockfield, LL.P
28 State Street
Boston, MA 02109

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
05/18/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/911,819	MICCO ET AL.	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 April 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3,5-32 and 34-55* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,5-32 and 34-55* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 4/17/07.

As indicated in Applicant's response, claims 1, 29 have been amended, and claims 4, 33 canceled. Claims 1-3, 5-32, 34-56 are pending in the office action.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-3, 5-7, 11-13, 17, 20, 29-32, 34-35, 39, 40-41, 45, and 48 are rejected under 35 U.S.C. 102(b) as being anticipated by Shannon et al., "Mapping the Interface Description Language Type Model in C", November 1989, IEEE Transactions on Software Engineering, Vol. 15, No. 11 (hereinafter Shannon).

As per claim 1, Shannon discloses a method comprising

providing a definition of a function associated with the first language about a function in a first language (e.g. … *existing languages ... LISP, Diana structures ... mapped ...to C, Ada Breadboard Compiler* – Introduction pg. 1333-1334; *...data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration ... function => name; String; parameters: Seq of ... formal parameters*, pg. 1334, right column; *class, function* - Fig. 1 pg. 1335; *Process declaration* - Fig. 2);

creating description information or *IDL* (*IDL specifications* - Fig. 3, pg. 1336; *IDL specification ... specifically, C macros ... and functions* – pg. 1334, left column, 4th para – Note:

The conversion based on a special package extending existing language structures to target C or Ada reads on creating a description interface on structures, e.g. a class definition, in first language, - into Ada function or C data structures; *data structures found in procedural programming languages* – 2nd para, right column, pg. 1334 – Note: the use of existing programming constructs reads on function in first language when data structure is inherent to function of a language);

translating a call to the function into a call to a corresponding function in a second language using the description information (e.g. *PLUM, ABC* -- V. Evaluation, Run-time efficiency, pg. 1342-1344; *...should be permitted by the C compiler* - Introduction pg. 1333-1334) without requiring processing of the definition of the function (e.g. *... should be natural and not require knowledge of implementation details* – pg. 1333, right column).

As per claim 2, Shannon discloses a file of description items and derived description information (e.g. sections II, III – pg. 1334-1339).

As per claim 3, Shannon discloses

examining the definition of the function associated with the first language (e.g. *...data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration ... function => name; String; parameters: Seq of ... formal parameters*, pg. 1334, right column – Note: the fact of creating appropriate parameters for a function disclose examining definition of a function in a source language);

and derive information (*IDL specifications* - Fig. 3, pg. 1336; *IDL specification ... specifically, C macros ... and functions* – pg. 1334, left column, 4th para – Note: generating of

IDL from existing functions written in some languages inherently teaches derive information for said functions) about the function.

As per claim 5, Shannon discloses creation of C language constructs, hence has implicitly disclose the *.lib* files associated with the assembling of object files prior to linking in C. Therefore, Shannon has implicitly disclosed library wherein entries associated with the assembling process in C compiler because at the time the invention was made it was a known concept that C compiler create lib files during a pre-linking compiler process.

As per claims 6 and 7, Shannon discloses a declaration with a set of input and output port (ch. B – pg. 1335; Fig. 2, pg. 1336) and input/output mapping (pg. 1338, Private types - right column); hence has disclosed analysis of the first language so to derive input/output formal parameters leading to creation of C formal parameters, i.e. list of input or output/return parameters otherwise the target function declaration would not result in a correct function declaration (Note: the declaration of formal input and output, and scope of variables declared in a function in 4th generation like C was a known concept at the time the invention was made; and according to C code translation by Shannon, this reads on input/outputs as in a formal declaration of a function as set forth in claims 1, 3).

As per claim 11, Shannon discloses a method in a computing device, comprising providing a file of description items, each including information about a function in a first language (e.g. *Diana structures ... mapped ...to C, Ada Breadboard Compiler* – Introduction pg. 1333-1334; *...data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration ... function => name; String; parameters: Seq of ... formal parameters*, pg. 1334, right column; *class, function* - Fig. 1 pg. 1335), wherein the description

enables translation of a call to the function in a first language into a call to a corresponding function in a second language (*...should be permitted by the C compiler* - Introduction pg. 1333-1334) without requiring processing of the definition of the function (e.g. ... *should be natural and not require knowledge of implementation details* – pg. 1333, right column); and

using information about a function associated with the first language to translate a first program file into a second program file (*...data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration ... function => name; String; parameters: Seq of ... formal parameters*, pg. 1334, right column; *class, function* - Fig. 1 pg. 1335 – Note: using existing programming language source to analyze structures in order to generate IDL constructs reads on using information about function in 1st language to translate a first program into a second program file – see pg. 1344 Runtime efficiency: *mapping, compiler*).

As per claims 12-13, referring to rationale of claims 6-7, Shannon has disclosed analysis of the first language input/output formal parameters leading to creation of C formal parameters, i.e. list of input or output/return parameters otherwise the target function declaration would not result in a correct function declaration (Note: the declaration of formal input and output, and scope of variables declared in a function in 4th generation like C was a known concept at the time the invention was made; and according to C code translation by Shannon, this reads on input/outputs as in a formal declaration of a function as set forth in claims 1, 3).

As per claim 17, Shannon teaches using existing programming language source to analyze structures in order to generate IDL constructs reads on using information about function in 1st language to translate a first program into a second program file – see pg. 1344 Runtime efficiency: *mapping, compiler)*; hence discloses for each call in the 1st program file, retrieving an

item from the file of description items, and using information description in the item to translate the first language function into a translated call corresponding to the 2nd language (Note: IDL and call declaration to functions by Shannon – refer to claim 11 -- read on retrieving item from description items; and using this information to translate into a corresponding function call in another programming language, a function call being translated reading on being stored after compiler translation).

As per claim 20, refer to rationale as set forth in claims 12-13.

As per claim 29, this is the computer-medium product version of claim 1, hence is rejected with the corresponding rejection as set forth therein.

As per claims 30-32, 34-35, these are the computer product claims corresponding to claims 2-3, 5-6; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claim 39, this is the computer-medium product version of claim 11, hence is rejected with the corresponding rejection as set forth therein.

As per claims 40-41, these are the computer product claims corresponding to claims 12-13; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claim 45, this is the computer-medium product version of claim 17, hence is rejected with the corresponding rejection as set forth therein.

As per claim 48, refer to rationale as set forth in claims 12-13.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 8-10, 14-16, 18-19, 36-38, 42-44 and 46-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shannon et al., "Mapping the Interface Description Language Type Model in C", November 1989, IEEE Transactions on Software Engineering, Vol. 15, No. 11, in view of Bjarne Stroustrup, "the C++ Programming Language", 2nd Edition, copyright 1991 (hereinafter Stroustrup).

As per claim 8, Namespace and function scope involving local and global parameters are known in 4th generation like C. Stroustrup discloses C programming language and namespace for addressing scope of members of a function dictated within a declaration directives of such namespace (see pg. 634, chp. 3.3.1). Hence, it would have been obvious for one skill in the art at the time the invention was made in light of Shannon type checking and preprocessing for runtime efficiency analyze a function scope while creating of description information about a function of the first language in order to derive a correct scope when effecting a C function declaration according to Stroustrup; because this would support the scope of definition of parameters by Shannon's C directives because this helps support scope of members within a class or function definition as endeavored by Shannon's preprocessing macro directives (see Shannon: pg. 1341-1344).

As per claims 9 and 10, Shannon does not explicitly teach determining of variable arguments in a function and a variable return of results. But according to Stroustrup's teaching of C++ providing a variable number of arguments, e.g. input arglist[], or argv[]; or a variable output/return in form of an array, or pointer to a struct or linked list (Stroustrup: *argv[]*– chp.

3.1.6, pg. 87; pg. 485 chp. 3.4; *argument ... list* - chp. 8.2.5 – pg. 532), this a known concept C construction at the time the invention was made. Hence, for one skill in the art at the time the invention was made, determining whether a function to have a variable input arguments or return variables would also have been obvious according the above teaching by Stroustrup to address possibility where number of arguments can vary, and analyzing and precisely controlling on such extensibility and variability of parameters as endeavored by Shannon' s type and code checking for runtime as set forth above would allow the target code to accommodate for such eventuality, using the known approach provided by C as mentioned above.

As per claims 14-16, refer to the rationale of claims 8-10.

As per claims 18-19, these claims subject matter fall under the ambit of the subject matter of claims 15-16; and are rejected using to rationale as set forth in claims 15-16, respectively.

As per claims 36-38, these are the computer product claims corresponding to claims 14-16; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claims 42-44, these are the computer product claims corresponding to claims 14-16; hence are rejected with the corresponding rejections as set forth therein respectively.

As per claims 46-47, these claims correspond to the subject matter of claims 15-16; and are rejected using to rationale as set forth in claims 15-16, respectively.

6. Claims 21-28, and 49-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Elmroth et al., "A Web Computing Environment for the SLICOT Library", December 2000, Brite-Euram III, Networks Programme NICONET, in view of Research Systems, "IDL",

copyright 1994, further in view of Shannon et al., "Mapping the Interface Description Language Type Model in C", November 1989, IEEE Transactions on Software Engineering.

As per claim 21, Elmroth discloses a method in a computing device comprising:

providing a library file including functions defined by a first language (e.g. *SLICOT Library, BLAS, LAPACK* – pg. 1, Introduction; *Riccati equations* - Fig. 2-3; *...uploaded ... Matlab files, data files, Latex, Scilab* – pg. 2, pg. 6, Fig. 1, 4 – Note: uploaded matrices in Matlab or Latex format , or Riccati equations read on library files - i.e. files served as input to a library generating tool - defined in one language, all such file integrated into the SLICOT library file framework);

creating a function library (e.g. *SLICOT Routines* – Fig. 6) and a description file (e.g. PHP scripts, Aux Routines – pg. 6-7 – Note: php scripts reads on being derived via parsing from the SLICOT and Aux Routines), the function library including one or more functions defined by a second language, each function in the function library being translated version of a function in the library file (e.g. *SLICOT routines* – pg. 6; *...written in C or Fortran* – pg.8: Conclusions and Future Work), and

using the description file to translate a program file from the first language into the second language (e.g. *SLICOT routines, Matlab binaries, PHP scripts* – pg. 7-8 – Note: library files xxxxMD or xxxOD files in conjunction with PHP scripts and converting math functions into m-files read on one or more functions translated from the library file in a second language, i.e. Matlab binaries function files)

But Elmroth does not explicitly disclose the description file including description information that enables translation of a call to the function in a first language into a call to a

corresponding function in a second language without requiring processing of the definition of the function; nor does Elmroth disclose that each call in the program file to a function in a first language is translated into a call to a corresponding function in the second language. Converting math functions into another program like high-level programming language like Fortran or C (analogous to suggestion by Elmroth – pg. 8: Conclusions and Future Work) was a known concept at the time the invention was made. Research_Systems, as set forth in claim 1, discloses interactive data language (IDL) and mapping various application functions to a fourth-generation programming language like C, and using such IDL file, i.e. description file similar to PHP scripts by Elmroth, to implement applications similar to the Riccati Equations or Matlab files by Elmroth, e.g. Math functions, graph plotting, Image Processing (see Research_Systems, pg. 1-5). The high-level definition description file by Research_Systems was an interfacing type of description language well known at the time the invention was made for converting functions in one language into a corresponding functions in another language, like from Math functions to C program as taught by Research_Systems. As set forth in claim 1, Shannon in a similar approach as Research_Systems, also discloses an interface description language (IDL) for mapping functions from one language to syntax and constructs implemented in C functions (re claim 1). Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide an IDL as taught by Reseach_Systems and Shannon to enhance the scripts by Elmroth so that the IDL description information is used instead of the PHP scripts, the IDL as to enable translation of a call to the function into a call to a corresponding function in a second language without requiring processing of the definition of the function; such that each call in the first language program file to a function in the library file is translated into a call to a

corresponding function in the second language (re claim 1: Shannon). The benefits of using IDL would have been obvious because of the same reasons listed by Shannon: the IDL provides type safe implementation into a high-level programming like C, and further does not require processing of the definition of the function (see Shannon: Introduction, pg. 1333-1334).

As per claim 22, this claim includes the limitation as to translate a call to the function in a first language into a call to a corresponding function in a second language as in claim 21; hence is rejected as set forth therein. Further, Elmroth discloses a translated version of each function in the library file (*SLICOT Library, BLAS, LAPACK* – pg. 1, Introduction; *Riccati equations* - Fig. 2-3; *...uploaded ... Matlab files, data files, Latex, Scilab* – pg. 2, pg. 6, Fig. 1, 4- Note: uploaded matrices in Matlab or Latex format , or Riccati equations read on functions in the library file - i.e. files served as input to a library generating tool - defined in one language, all such file integrated into the SLICOT library file framework).

As per claim 23, this claim limitation corresponds to that of claim 3; hence is (in view of the combination Elmroth and Research_Systems and Shannon from above) rejected as obvious using most of Shannon teachings, mapping of function using a description file derived from examining a function in the first language library file, in light of Research_Systems.

As per claims 24 and 25, these claims correspond to limitations of claim 3, 4 and 11; hence are rejected using the corresponding Shannon's teachings in view of the rationale to combine Elmroth, Research_Systems and Shannon as set forth above.

As per claim 26-28, Elmroth discloses a Web call mapping and converting interface (Fig. 6), while Shannon discloses an interface to check call for error and type violation (the User Interface – pg. 1344). In light of the rationale as to combine the IDL teachings by Shannon with

Elmroth's web interface and PHP scripts, the motivation to provide Elmroth' s Web interface a function evaluation interface as suggested by Shannon would have been for the same reasons as combining Elmroth with Shannon as in claim 21. Further, Elmroth does not specify variable input descriptor, variable output descriptor, descriptor for a known number of input or output arguments as recited in claims 18-20. In view of the rationale to combine Elmroth with the IDL by Research_Systems and Shannon, these claims will be rejected as in claims 18-20 respectively.

As per claim 49, this is the computer-medium product version of claim 21, hence is rejected with the corresponding rejection as set forth therein.

As per claims 50-56, these are the computer product claims corresponding to claims 22-28; hence are rejected with the corresponding rejections as set forth therein respectively.

Response to Arguments

7. Applicant's arguments filed 4/17/07 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

35 USC § 102 Rejection:

(A) Applicants have submitted that, for claim 1, Shannon's IDL for describing data structures specifications to map with C data structures does not disclose translating a call to the function in the first language into a call to a corresponding one in a second language (Appl. Rmrks pg. 13). This appears to be a repeat of a previously presented argument. In reply thereto, a counter argument has set forth as following.

It is noted that the IDL tool is to provide all the necessary description information enabling a target language programming constructs to be converted therefrom. Because the claim does not recite more specifics as to enable one skill in the art to acquire adequate

understanding behind the concept of 'translating a call' into a call (to a corresponding function), such call translation has been interpreted broadly as a translation of a function for a runtime using a IDL definition by Shannon, i.e. effect code constructs in a another programming language in Shannon runtime. Thus the linking as set forth in the rejection reads on function calls translated from IDL definition based on a first language constructs (see Shannon: *...data structures found in procedural programming languages* – 2nd para, right column, pg. 1334; *node declaration ... function => name; String; parameters: Seq of ... formal parameters*, pg. 1334, right column; *class, function* - Fig. 1 pg. 1335); hence Shannon has disclosed such translation of a call via translation of a function for a runtime using said IDL definition of a function call.

For one of ordinary skill in the art of compiling, **translating a call from a first language into a second language** is the process by which parameters passing/returning and function call implementation is effectuated in terms of internal program constructs conversion into a target language via mapping from one source language syntax into that target language syntax such that this syntactically converted function would be able to execute (as a call) when compiled in its executable form. The claim language does not provide any particular implementation details that would enforce another way of interpreting the above concept. The above translating as construed has been deemed equivalent to what Shannon teaches; that is, providing an intermediate definition file to map into a target second language from any first language such that after being converted into the second language syntax, a procedure call can be executed, i.e. a call translation. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

The broadness of the recited 'translating' step cannot obviate or preclude the manner by which the IDL support enables Shannon to translate a first language function into its corresponding function in a second language; and a translation into such function and a translation of a procedure call is the same. The argument is largely unconvincing.

(B) Applicants have submitted that Shannon's IDL declaration of structures is not description information **about a function** as recited I claim 11 (Appl. Rmrks pg. 14, top). The rejection has provided portion that show an IDL with information about how a procedure internal constructs in terms of its parameters as defined. Applicants also disagree that a second language does not necessarily mean C language; however, a second language being C is still anticipating what is claimed because the IDL is purported to do the mapping from some first language (refer to rejection) that is not the same as the C language being the target language. The argument is not persuasive.

(C) Applicants have submitted that claims 29-35, 39-41, 45 and 48 have not been anticipated because Shannon does not disclose 'description information about the function ... and translate a call ... using the description information (Appl. Rmrks pg. 15). These arguments will have to be referred to sections A and B above for they are mere allegation for patentability that appears to have been reading more into the claims than broad language of the claims entails.

35 USC § 103 Rejection:

(D) Applicants have submitted that for claims 8-10, the combination of Shannon and Stroustrup does not disclose or suggest 'creating description information about a function from the definition of a function... first language'; and that Stroustrup's character strings as in argv would not be same as 'description information ... function ... first language' (Appl. Rmrks pg.

16). The rationale as set forth in the 103 rejection is not meant to address 'description information ... about a function ... first language' nor is it for using Stroustrup to anticipate a feature that Shannon already teaches. The argument appears to bear on the feature that has been addressed using § 102 rejection (i.e. expounding on this feature by approaching Shannon and Stroustrup as this feature were a separate anticipation type of rejection); and therefore not directly commensurate with the grounds of rejection as set forth in the 103 rationale. Besides, in response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

(E) Applicants have submitted that claims 14-16, 18-19, 36-38, 42-44, 46-47 depend on a claim or claims that have not been anticipated because Shannon (in view of Stroustrup) does not disclose or suggest creating or providing 'description information about the function ... and translate a call ... using the description information'(Appl. Rmrks pg. 17). These arguments will have to be referred to sections A and B above for they are mere allegation for patentability that appears to have been reading more into the claims than broad language of the claims entails.

(F) Applicants have submitted that for claim 21, Elmroth 's teaching of SLICOT with an ability to download file in different formats is not 'creating a function library and description file including description information about each function ... library file' wherein the 'description information ... call to the function ... second language' (Appl. Rmrks pg. 18). In reply, the rejection has set forth the approach of converting Math functions by Elmroth into C language; and why a definition file to support conversion would be lead to an obvious feature that one of

ordinary skill would effectuate, based on the similarity of Research Systems by Shannon and Elmroth. That is, textually:

Converting math functions into another program like high-level programming language like Fortran or C (analogous to suggestion by Elmroth – pg. 8: Conclusions and Future Work) was a known concept at the time the invention was made. Research_Systems, as set forth in claim 1, discloses interactive data language (IDL) and mapping various application functions to a fourth-generation programming language like C, and using such IDL file, i.e. description file similar to PHP scripts by Elmroth, to implement applications similar to the Riccati Equations or Matlab files by Elmroth, e.g. Math functions, graph plotting, Image Processing (see Research_Systems, pg. 1-5). The high-level definition description file by Research_Systems was an interfacing type of description language well known at the time the invention was made for converting functions in one language into a corresponding functions in another language, like from Math functions to C program as taught by Research_Systems. As set forth in claim 1, Shannon in a similar approach as Research_Systems, also discloses an interface description language (IDL) for mapping functions from one language to syntax and constructs implemented in C functions (re claim 1)...

There is no mentioning in the above rationale about ability to download by Elmroth being Elmroth's anticipating the recited function library and *description file* as asserted above. The rejection has also mentioned that IDL by Research_Systems is 'interactive data language' along with the 'interface description language' by Shannon. However, the language of the claim amounts to 'description file' wherein 'description information enables translation of a call to the function in the first language into a call to a corresponding function in the second language'; and

this *description file* as understood and construed cannot enforce mapping using a particular IDL so that it can only bear one name and not another. That this *description file* is construed via names being 'interactive description language' or 'Interactive Data Language' is irrelevant because its patentable weight is only perceived via its (recited) functionality; which the rejection is based upon to address. Accordingly, the rejection has been explicit in mentioning that IDL by Research_Systems is also as **descriptive** as the IDL by Shannon in terms of providing descriptive information based upon which to translate on function in one language to another language. The argument that 'Interface description Language' and 'Interactive Data Language' amount to an Examiner's mix-up of concepts is therefore misplaced. The claim language in view of the way it has been interpreted (refer to section A and B) has been at the foundation by which the rationale of the 103 rejection has been effectuated. Again, the Applicants contend with alleging that Elmroth does not have the description file; which is not commensurate with the above 103 rationale which goes from identifying the missing feature, explaining the analogous endeavors in the references and combine the teachings so the meet the claim as a whole in a obviousness rationale. Applicant seem silent in detecting how the combination as set forth would have yielded negative effects or that how combining Elmroth's approach with Shannon' s IDL or Research_Systems's interface definition language file would be inappropriate. Until the argument successfully point out in which such combination would result in incongruous and unwanted outcome, the argument is deemed insufficient to overcome the rejection. The arguments are therefore insufficient to overcome the rejection.

(G) Applicants have submitted that claims 22-28, 49-56 are allowable based on the failure of Shannon and the references as set forth above. These arguments fall under the ambit of the sections above to address the same and repeated grounds raised by these arguments.

The claims will stand rejected as set forth in the Office Action.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Tuan A Vu
Patent Examiner,
Art Unit 2193
May 09, 2007